

**SYSTEM AND METHOD FOR THE CREATION AND AUTOMATIC DEPLOYMENT
OF PERSONALIZED, DYNAMIC AND INTERACTIVE VOICE SERVICES, WITH
SYSTEM AND METHOD THAT ENABLE ON-THE-FLY CONTENT AND SPEECH
GENERATION**

Field of the Invention

This invention relates to a system and method for creation and automatic deployment of personalized, dynamic and interactive voice services, including information derived from on-line analytical processing (OLAP) systems, where the
5 system and method include a system and method that enable on the fly content and speech generation.

Background of the Invention

Interactive telephone systems enable a user to interactively request information
10 through a computerized interface. These systems typically utilize predefined or prerecorded menus that a user accesses by calling in to a central number. The predefined menus typically enable a user to request information by stepping through various predefined choices that remain the same regardless of who the caller is or of the content of the information being accessed. Such information may include accessing account
15 information, movie times, service requests, etc.

A problem with these systems is that the menu structure is typically set and not customized to a particular's users preferences or customized to the information available

to that user. Therefore, a user may have to wade through a host of inapplicable options to get to the one or two options applicable to that user. Further, a user may be interested in particular information. With existing telephone call-in systems, that user has to input the same series of options each time they want to hear that information. If the user desires to
5 hear the information frequently, the telephone input system described is a very time consuming and wasteful method of accessing that information. Moreover, if a user is interested only in knowing if a particular value or set of values has changed over a predetermined period of time, in such a system, the user is required to initiate a call and wade through all of the information to determine if the particular value has changed.

10 These and other drawbacks exist with current systems.

Summary of the Invention

An object of the invention is to overcome these and other drawbacks in existing systems.

According to one embodiment, the invention provides a system and method for
15 creation and automatic deployment of personalized, dynamic and interactive voice services, including information derived from on-line analytical processing (OLAP) systems, where the system and method include a system and method that enable on the fly content and speech generation.

One embodiment of the invention relates to a system and method for creation and
20 automatic deployment of personalized, dynamic and interactive voice services, including information derived from on-line analytical processing (OLAP) systems and other data

repositories. The system and method enables the ability to capture user selections to facilitate closed-loop transaction processing and processing of other requests. One aspect of the invention relates to an interactive voice broadcasting system and method that enables analytical reporting and advanced transactional services via the telephone or other voice-enabled terminal device. One advantage of the invention is that a voice service may leverage the power of OLAP or other data repository systems and provide critical information to the user, in a timely fashion, by phone. Another advantage of this method and system is that it provides a user with the opportunity to immediately act upon information received during a interactive voice broadcast.

10 A voice service is created and can have many users subscribed to the voice service. Each user can specify personal preferences for the content and presentation of the contents for a voice service. The specification of the elements of a voice service may be done using a set of interfaces (such as GUIs) that take the form of a voice service wizard.

15 A voice service includes one or more Dialog elements. Dialog elements may include one or more of Speech elements, Input elements and Error elements. An Input element may include a Prompt element and/or an Option element. An Input element enables the system to request input from the user, capture the input and direct the call flow based on the user's input. An Option element associates a key (e.g., on a telephone touch pad dial) with a destination Dialog that is executed when that number is pressed by
20 a user during an interactive voice broadcast. A Prompt requests a user to enter numeric or

other information. An Input element may enable a user to request, during an interactive voice broadcast, a transaction, a service or other requests. The term transactions, services and requests are to be interpreted broadly.

According to one embodiment, the user's responses to Input elements are stored
5 during an interactive voice broadcast and, during or after the voice broadcast, the stored information is processed by the system or is passed to another system or application for processing. The transaction (or other request) processing can be accomplished either in real-time, during the voice broadcast, or after the interactive voice broadcast is completed. The results or confirmation of a transaction or other request can be provided
10 to the user during the call or subsequently.

Once a voice service is created, the system monitors predetermined conditions to determine when the voice service should be executed. Each voice service is executed when one or more predetermined conditions are met as specified during creation of the voice service. For example, a voice service may be executed according to a
15 predetermined schedule (time-based) or based on a triggering event (e.g. one or more conditions are met based on the output of an OLAP or other report).

When the predetermined condition is satisfied, the voice service is executed. Executing a voice service, includes the steps of generating the content specified by the voice service and the user preferences. Some users may have identical personalization
20 options and, thus, a single call structure may be generated for a group of users with identical personalization options. The content of the voice service includes the

information that is to delivered to users of that voice service, and the Input to be requested from the user, among other things. The content may include, for example, static text messages, dynamic content (e.g. text based on information output from an OLAP report, other database or other sources), blended text (e.g. static text combined
5 with dynamic content) and prerecorded sound files.

This and other content along with a users personalization preferences are formatted in an Active Voice Page (AVP). An AVP contains the call structure and data, voice style parameters for the user and personal identification information designated for the user. The AVP contains data at various hierarchical levels that are defined by the
10 Dialog elements defined for each voice service. The active voice pages are used to help govern the interaction between the call server and the user during an IVB. According to one embodiment, the content is formatted, into an AVP using XSL stylesheets so the AVP is in an XML-based language. According to one embodiment, the XML-based language used is a novel language referred to as TML (discussed below). Other XML-
15 based markups could be used such as VoiceXML™. The AVP is sent to a call server along with style properties for each user. The style properties of a user help determine the behavior of the call server during an interactive voice broadcast. A unique AVP is generated for each user scheduled to receive a voice service.

When a user is called by the call server, information is passed through a T-T-S
20 engine and delivered to the user through a voice-enabled terminal device. Preferably, the structure of each call is dynamic, driven by current data values and is personalized based

on a user profile established during subscription to a voice service. During a typical interactive voice broadcast, a synthesized, natural sounding voice greets the recipient by name, identifies itself, provides information relevant to the user and enables a user to provide input back to the system.

5 An IVB is a voice-enabled interaction with a user having a dynamic structure controlled by the AVP for the particular user. The IVB may be delivered using real-time, on-the-fly speech generation. During an IVB, information is exchanged between the call server and a user according to the AVP. The system executes dialogs by reading messages to the user and, eliciting input from the user. For example, the user may press
10 buttons on a telephone touch pad dial to select an option or to provide numeric or alphanumeric input or the user may speak a response which the system resolves using speech recognition technology. Each response provided by a user may transfer control of the IVB to a different part of the AVP or to a different AVP.

 During or after the IVB, the user's responses may be processed by the system or
15 other applications. The AVP may contain pointers to other applications and embedded statements such that when a user exercises an option, the system performs a requested operation and returns the results to the user during the IVB. For example, by exercising an option, a user may request that a real-time database query be performed. When the user selects such an option, control is shifted to a portion of the AVP that contains an
20 embedded SQL statement that is made against a database.

When a user has worked through selected dialogs of the AVP, the IVB is terminated. That is, a user likely will not work through all of the available dialogs during an IVB. Rather, the user's inputs and option selections determine which the available dialogs are encountered during any given IVB.

5 Other features and advantages of the present invention will be apparent to one of ordinary skill in the art upon reviewing the detailed description of the present invention.

Brief Description of the Drawings

Fig. 1a is a flow chart of a method in accordance with an embodiment of the
10 present invention.

Fig. 1b is a flow chart indicating a method of generating a voice service according to one embodiment of the present invention.

Fig. 1c is a flow chart indicating a method for interactive voice broadcasting according to an embodiment of the present invention.

15 Fig. 2 is a flow chart indicating a sequence of an interactive voice broadcast according to one embodiment of the present invention.

Fig. 3a is a schematic block diagram of a system in accordance with an embodiment of the present invention.

Fig. 3b is a schematic block diagram of an intelligence server according to an
20 embodiment of the present invention.

Fig. 3c is a schematic block diagram of call server according to an embodiment of the present invention.

Fig. 4 is a schematic block diagram of a commercial transaction processing system according to an embodiment of the present invention.

5 Fig. 5 is a flow chart of a method of using a voice service bureau according to an embodiment of the present invention.

Fig. 6a is a schematic block diagram of a voice service system incorporating a voice service bureau according to one embodiment of the present invention.

10 Fig. 6b is block diagram of a primary voice bureau according to one embodiment of the present invention.

Fig. 6c is a block diagram of a backup voice bureau according to another embodiment of the present invention.

Fig. 7 is a flow chart illustrating a method for integrating inbound and outbound voice services.

15 Fig. 8 is a block diagram of a call server configured to provide integrated inbound and outbound voice services.

Fig. 9 is a flow chart of a method for on-the-fly generation of voice menus used to drive an interactive voice broadcast according to one embodiment of the present invention.

Detailed Description of the Preferred Embodiments

The method and system of the present invention enables on the fly generation of voice menus that are used to drive an IVB. According to one embodiment, the voice menus are generated from a unique markup language created to facilitate interactive voice broadcasting. According to one embodiment, the markup language used is TML. Other XML-based markups, such as VoiceXML™ could be used.

To facilitate understanding of the system and method of the present invention, a brief explanation of TML is provided. TML is a markup language that enables interactive voice broadcasting. According to one embodiment, TML is based on XML and comprises a set of elements that are used to define functions for various portions of a document and a set of tags that correspond to the elements. The tags are used to delimit portions of a document that belong to a particular element.

According to one embodiment, TML comprises a Header Element, a Container Element, a Dialog Element, a Text Element, a For-Each Element, an Input Element, an Option Element, a Prompt Element, an Error Element, and a System-Error Element.

A Header Element is used to identify the markup language on which a document is based.

A Container Element is used to identify a document as a TML document.

A Dialog Element is the basic unit of interaction between a system and a user.

According to one embodiment, a Dialog Element may contain text that is to be spoken to

a user. A Dialog Element may contain Text Elements, For-Each Elements, Input Elements, Option Elements and Prompt Elements.

A Text Element may also be called a Speech Element and defines text portions of a document. According to one embodiment, text portions of a document are used to
5 specify information that is to be spoken to a user.

A For-Each Element is used to cycle (loop) through a group of related variables. to dynamically generate speech from data.

An Input Element defines sections of Dialog Elements that contain interactive portions of the TML document. According to one embodiment, an Input Element
10 contains elements that pertain to a response expected from a user.

An Option Element identifies a predefined user selection that is associated with a particular input. According to one embodiment, Option Elements are used to associate one or more choices available to a user with telephone keys.

A Prompt Element defines a particular input that is expected. According to one
15 embodiment, a Prompt Element defines that a sequence or number of key presses from a telephone keypad is expected as input. Unlike an Option Element, a Prompt Element is not associated with predefined user selections.

An Error Element defines a response to invalid input by a user. For example, an Error Element may be used to define the response to entry of an undefined option.

A System-Error Element defines a response to predetermined system events. For example, a System-Error Element may be used to define the response to expiration of the waiting time for a user input.

Within any given document, TML uses a set of corresponding tags to delimit each
5 of the above defined elements.

According to another embodiment, a TML document is a collection of the above described elements. Within a document, the boundaries of an element are delimited by its corresponding tags. Moreover, according to one embodiment, elements are arranged as parent elements and child elements. Parent elements may contain text and other
10 elements. If an element is contained by another element, it may be called a child of the containing element.

According to another embodiment, a TML document is used to provide interactive, dynamic voice services to a user through a telephone or other voice-enabled terminal device. A TML document enables a user to receive dynamically generated
15 information and provide various types of input in response. According to one embodiment, the TML elements and tags described above are used to specify text that is to be communicated to a user and to request input from a user. According to this embodiment, the specified text is passed through a text-to-speech converter and conveyed to a user over a telephone.

20 According to one embodiment, Dialog elements identify the portions of the TML document that communicate with a user. Within a Dialog element, Text Elements and

For-Each Elements define text that is to be read to a user. Input Elements identify the portion of a Dialog Element that are interactive with a user. Within an Input Element, Option Elements and Prompt Elements may define text to be read to a user, but they also request input from a user. According to one embodiment, one or more Option Elements
5 may include text that requests that a user choose one or more items from a list of choices defined by the Option Elements using the telephone keypad or by speaking a response. According to another embodiment, a Prompt Element may include text that requests free-form input from a user, e.g., by entering alpha-numeric characters using the telephone keypad or speaking a response.

10 With respect to the use of spoken responses, according to one embodiment, speech recognition technology is used to enable a user to respond to a prompt element or to select an option element verbally by saying a number, e.g., "one.". The verbal response is recognized and used just as a keypress would be used. According to another embodiment, the user may provide a free form verbal input. For example, a prompt
15 element may request that a user enter, e.g., the name of a business. In response the user speaks the name of a business. That spoken name is then resolved against predetermined standards to arrive at the input. Word spotting and slot filling may also be used in conjunction with such a prompt to determine the user input. For example, a prompt may request that the user speak a date and time, e.g., to choose an airline flight or to make a
20 restaurant reservation. The user's spoken response may be resolved against known date and time formats to determine the input. According to another embodiment, a prompt is

used to request input using natural language. For instance, in conjunction with a voice service to be used to make travel plans, instead of having separate prompt elements request input for flight arrival, departure dates and locations, a single natural language prompt may ask, "Please state your travel plan." In response, the user states 'I'd like to
5 go from Washington DC to New York city on the 3rd of January and return on the 3rd of February. This request would be processed using speech recognition and pattern matching technology to derive the user's input.

The TML document may comprise Error element and System-Error Elements. According to one embodiment, an Error element includes text that notifies a user of an
10 invalid input. The System-Error element may also include text that notifies a user that the system has experienced an undefined event, e.g., a non-response to an Input Element.

Figure 9 is a flow chart showing a method enabling real-time content and speech generation in accordance with one embodiment of the present invention. The method of Figure 9 begins with initiation of a call (step 901). According to one embodiment, a call
15 is initiated by reading a record from a call process table in a database. According to one particular embodiment, when the method of Figure 9 is used in conjunction with the system shown in Figures 1-8, a call is initiated when call server 18 reads a call string from call database 1811. According to another embodiment, a call is initiated by an incoming call. Standard telephonic hardware and software may be used to detect an
20 incoming call. When the method of Figure 9 operates in conjunction with the system

shown in Figures 1-8, an incoming call may be received on phone lines 183 and a caller verified as described below.

A TML document for the call is accessed in step 902. According to one embodiment, when operating in conjunction with the system of Figures 1-8 a TML document is stored in call database 1811 and sent to XML parsing engine 1812 when the
5 call is initiated. According to another embodiment, when an incoming call is received, the TML document is accessed from call database 1811 using a simple search as described below in conjunction with Figures 1-8.

In step 903, the TML document is validated. According to one embodiment, the
10 validation of the TML document comprises determining whether or not the TML document has proper beginning and ending TML tags. That is, validation of the TML document comprises determining whether or not the TML document is complete. According to one particular embodiment, when the method of Figure 9 is used in conjunction with the system shown in Figures 1-8, TML validation is accomplished by
15 TML parsing engine 1812 of call server 18.

A call channel is created for the call in step 904. According to one embodiment, creating a call channel comprises among other things, establishing a physical call and a response collection to maintain state and variable information for the duration of the call. For example, during an IVB a user may provide responses to particular menu options
20 given. Those responses are stored in variables and maintained throughout the call as described in more detail below. According to one embodiment, establishing the physical

call comprises establishing an actual connection between the user and the voice service system. According to one particular embodiment, when the method of Figure 9 operates in conjunction with the system of Figures 1-8, the response collection is stored in user response module 1815 and a connection is established using phone lines 183.

5 In step 905, a determination is made whether the voice service system is connected to a person or a device. According to one embodiment, this determination is made using a simple user prompt. If no response to the prompt is received it can be determined that the system is connected to a device. If a response is received, it can be determined that the system is connected to a person. According to one particular
10 embodiment, when the method of Figure 9 is used in conjunction with the system of Figures 1-8, the device/person determination is made using a software module in call server 18. Other embodiments are possible.

If the system is connected to a device, the method proceeds to step 906. In step 906, text of a message to be left on the device is accessed. According to one
15 embodiment, when the method of Figure 9 operates in conjunction with the system of Figures 1-8, text is accessed using a call to TML engine 1812 to "GetNextStep" along with the argument "device." This request to TML engine 1812 directs control to a portion of the TML document containing the text of a message to be left on the user's device. According to one embodiment, the message may comprise a notification that the
20 user has received a voice service and a number that the user may call to access that voice service.

In step 907, the text that is accessed is spoken to the device. According to one embodiment, the text of the message to be left for the user is converted to speech and delivered to the device. According to one particular embodiment, when the method of Figure 9 is used in conjunction with the system of Figures 1-8, the text of the message is forwarded to text to speech engine 1814 and converted to speech in real-time as described below.

After the message is spoken to the user's device in step 907, the call channel is terminated in Step 908. According to one embodiment, when a call is received by a device, there are no user responses to process and thus the method of Figure 9 terminates.

Returning to step 905, if it is determined that the voice service system is connected to a person, text to be spoken to the person is accessed in step 909. According to one embodiment, when the method of Figure 9 is operating in conjunction with the system of Figures 1-8, text is accessed through a command to TML parsing engine 1812 to "GetNextStep" along with the argument "person." According to one embodiment, such a command directs control of the TML parsing engine to a portion of the TML document that contains content to be read to the user. Other command structures could be used.

In step 910, a command is also accessed. According to one embodiment, the command comprises instructions on what to do after the text accessed in step 909 is delivered to the user. According to various embodiments, the command may comprise a command to terminate the call, a command to wait for a single digit input from the user,

or a command to wait for a multiple digit input from the user. Other commands are possible. According to one particular embodiment, the command that is accessed is part of the TML document that controls the IVB.

In step 911, the message to be spoken to a user is generated and delivered.

5 According to one embodiment, the text accessed in step 909 is converted to speech and delivered to a user. According to one particular embodiment, when the method of Figure 9 is used in conjunction with the system described in Figures 1-8, the text accessed in step 909 is sent to text-to-speech engine 1814 and converted to speech that is delivered to a user through the established call channel.

10 According to one embodiment, the text is generated and delivered in a manner dictated by style properties selected for the user. According to an alternative embodiment, the manner in which the text is generated and delivered to a user may be selected or altered during the IVB by the user. For example, the user may have the option of choosing the language styles (e.g., the French/English, female/male) during the during
15 the IVB. As another example the user may choose short or long forms of menus ("If you would like to hear more information press 9" vs. "More information, press 9") during the IVB.

In step 912, a determination is made whether or not user input is expected. According to one embodiment, this determination is made based on the command
20 accessed in step 910. According to this embodiment, based on the TML document, the command determines what is to be done after delivering text to a user. For example, if

the text to be delivered to a user was part of a Prompt element or an Option element, the voice service system expects a response from the user.

If user input is expected control passes to step 913 where user input is received. According to one embodiment, user input is received at using standard telephony technology to interpret a user's keypad presses. According to another embodiment, 5 speech recognition software is used to receive and interpret a user's spoken response. According to a particular embodiment, user responses are received at call builder 1813 through telephone lines 183.

It is determined whether or not the user input is in the proper form in step 914. As 10 mentioned above in conjunction with step 910, user input may comprise a single digit or may comprise multiple digits or other forms of input depending on the dialog from which it originated. According to one embodiment, in step 914, when it is determined whether or not a user input is in the proper form, the user input is checked against the criteria specified in the TML element that defined it. At the beginning of step 909, TML engine 15 1812 has the internal variables for the current call, the current position in the TML document, and the last user input (or the "person" or "device" argument if control is at the beginning of the call). Thus, the TML engine 1812 knows what input to expect from a user because that type of input is specified in the TML element that generated the text and the command. For example, the TML element <PROMPT type="fixed" length="3" 20 filter="\n gt 50 and \n lt 150" next="main"/> accepts a three digit number (Type = "fixed", Length = "3") that falls in the range between 50 and 150. If a user entered a

number that meets these criteria, this PROMPT element will be executed. Other command structures are possible.

If user input is not in the proper form, an error message is returned in step 915. According to one embodiment, step 915 comprises accessing a portion of the TML document that contains an error message to be delivered to the user. In particular, if it is determined that the user has entered an improper input form, a command "GetNextStep" along with the argument error is returned to XML parsing engine 1812. In response to this command, XML parsing engine accesses a portion of the TML document which contains a response notifying the user that they have entered an erroneous form of input. The user may then be prompted to enter the proper input.

If in step 914 it is determined that the user input has taken the proper form control is passed to a new dialog element. According to one embodiment, the "GetNextStep" with the argument "next" is used to transfer control to a different dialog element. This dialog element could be in the same TML document, or in a different TML document. If the TML element is in a different TML document, the full URL of the TML document is specified. Other command structures are possible.

When the next dialog element has been determined, control is passed to step 909 where the next text to be spoken to the user is accessed. Similar to that described above, according to one embodiment, the next text to be spoken to the user is accessed using a command "GetNextStep" with the argument that comprises the user response. Thus, the

user response acts as a pointer, to a new portion of the TML document containing the next text to be spoken to the user.

TML elements that request user input (i.e., Prompts and Options) may also include "store" and "remove" attributes that update the response collection. According to one embodiment, a "store" attribute indicates that the user response should be added to the response collection and the "remove" attribute indicates that a particular variable will be removed if a certain response is received. Other attributes are possible. Thus, after it is confirmed that an input of the appropriate form has been received, any "store" "remove" or other attributes are executed and result in changes of the variables.

Returning to step 912, if it is determined that no user input is expected, control passes to step 908 where the logical call channel is terminated. According to one embodiment, the logical call channel is terminated by breaking the connection with the user.

In step 916, any user response returned during the course of the call is stored for processing. According to one particular embodiment, responses received during the call are processed as discussed in conjunction with Figures 1-8. According to another embodiment, user responses may be processed during the course of the call. According to this embodiment, after user input has been received in step 912 and verified in step 913, the user response is forwarded to the appropriate module for processing. According to one embodiment as described in conjunction with Figures 1-8, the user response may be used to generate a database query against database system 12. According to another

embodiment as described in conjunction with Figures 1-8, the user response may be used to complete a real-time e-commerce transaction. According to these embodiments, when the transaction is completed, control is returned to step 909 and additional text to be spoken to a user is accessed.

5 The system and method described above may be used in conjunction with an overall voice service method and system described above. According to one embodiment of the present invention, a system is provided for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers. For example, subscribers may be called by the system, and have content delivered audibly over the telephone or other
10 voice-enabled terminal device. During the IVB, information may be exchanged between the system and a subscriber. The system conveys content to the subscriber and, the subscriber may respond by pressing one or more buttons on a telephone touch pad dial (or other input mechanism) to hear more information, to exercise options, or to provide other responses. This interaction shapes the structure of a basic exchange between the system
15 and the subscriber. During or after the call is terminated, the subscriber's responses may be stored and processed (*e.g.*, by other applications).

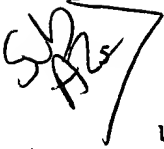
 According to one embodiment of the present invention, a method for automatic, interactive, real-time, voice transmission of OLAP output to one or more subscribers is provided. Figure 1a depicts a flow chart of a method for automatic, interactive, real-time,
20 voice transmission of OLAP output according to one embodiment. The method begins in step 110 with the creation of a voice service (*e.g.*, by a system administrator or user). A

voice service is created using, for example, a voice service wizard which may comprise a series of interfaces. One embodiment of a method for creating a voice service is explained in more detail below in conjunction with Figure 1b. One embodiment of a voice service wizard is explained below in conjunction with Figure 3b.

5 After a voice service is created, users may subscribe or be subscribed to the voice service (step 120), for example, by using a subscription interface. According to one embodiment, users may subscribe to an existing voice service over the telephone or by web-based subscription. A user may also be subscribed programmatically. In other embodiments, a user may subscribe to a voice service via electronic mail. Not every
10 voice service created in step 110 is available for subscription. More specifically, according to another embodiment, only a user with appropriate access, such as the creator of the service, is allowed to subscribe himself or others to a service. Such a security feature may be set when the voice service is created.

 In step 130, a scheduling condition or other predetermined condition for the voice
15 services is monitored to determine when they are to be executed. That is, when a voice service is created or subscribed to, the creator or user specifies when the voice service is to be executed. A user may schedule a voice service to execute according to the date, the time of day, the day of the week, etc. and thus, the scheduling condition will be a date, a time, or a day of the week, either one time or on a recurring basis. In the case of an alert
20 service, discussed in more detail below, the scheduling condition will depend on satisfaction of one or more conditions. According to one embodiment, the condition(s) to

be satisfied is an additional scheduling condition. According to another embodiment, to another embodiment, a service may be executed "on command" either through an administrator or programmatically through an API. Scheduling of voice services is discussed in more detail below.

 The method continues monitoring the scheduling condition for voice services until a scheduling condition is met. When a scheduling condition is met, that voice service is executed. The execution of a voice service involves, inter alia, generating the content for the voice service, and structuring the voice service to be interactively broadcast through a call server. The execution of a voice service is explained in detail in
10 conjunction with Figure 1c.

An example of an IVB is as follows.

Hello Joe, this is your stock update.

PIN VERIFICATION

5 Please enter your six digit PIN number

(Joe enters his PIN, using the keypad dial on his telephone.)

MENU OPTIONS

Your portfolio was up by \$1000 today.

10 Please select:

1. To get a portfolio stock update
2. To conduct a transaction

(Joe presses 2)

15 SUB MENU

Thank you, Joe! Please select a ticker.

1. PQT
2. TQP
3. Listen to options again
4. Return to main menu

(Joe presses 1.)

SUB MENU

Would you like to buy or sell stock? Please press:

1. To sell stock
2. To buy stock

(Joe presses 1.)

SUB MENU

How many shares of PQT would you like to sell today? Please press:

1. To sell 50 shares
2. To sell 100 shares
3. To sell 200 shares
4. To sell another quantity

(Joe presses 2.)

SUB MENU

You selected 2. You want to sell 100 shares of PQT. Please press:

1. If this is correct
2. If this is incorrect
3. If you want to change the number of shares you want to buy.

(Joe presses 1.)

END VOICE SERVICE/TERMINATE IVB

Thank you for using our voice interactive broadcasting service, Joe. We
will call you

5 back when your transaction is completed. Good-bye.

As can be seen from the above sample during an IVB, the user is presented with
information, *e.g.*, the status of his portfolio, and is presented options related to that report,
e.g., the option to buy or sell stock.

10 According to one embodiment, a voice service is constructed using service
wizard. A voice service is constructed using several basic building blocks, or elements,
to organize the content and structure of a call. According to one embodiment, the
building blocks of a voice service comprise elements of a markup language. According
to one particular embodiment, elements of a novel markup language based on XML
15 (TML) are used to construct voice services. Before explaining how an IVB is
constructed, it will be helpful to define these elements.

The DIALOG element is used to define a unit of interaction between the user and
the system and it typically contains one or more of the other elements. A DIALOG can
not be contained in another element.

20 The SPEECH element is used to define text to be read to a user.

The INPUT element is used to define a section of a DIALOG that contains interactive elements, *i.e.*, those elements that relate to a response expected from a user and its validation. An INPUT element may contain OPTION, PROMPT and ERROR elements.

5 An OPTION element identifies a predefined user selection that is associated with a particular input. According to one embodiment, OPTION elements are used to associate one or more choices available to a user with telephone keys.

 A PROMPT element defines a particular input that is expected. According to one embodiment, a PROMPT element defines that a sequence or number of key presses from
10 a telephone keypad is expected as input. Unlike an OPTION Element, a PROMPT Element is not associated with predefined user selections.

 The PROMPT and OPTION elements may also be used to request user input using natural language. According to one embodiment, speech recognition technology is used to enable a user to respond to a PROMPT element or to select an OPTION element
15 verbally by saying a number, e.g., "one.". The verbal response is recognized and used just as a keypress would be used. According to another embodiment, the user may provide a free form verbal input. For example, a PROMPT element may request that a user enter, e.g., the name of a business. In response the user speaks the name of a business. That spoken name is then resolved against predetermined standards to arrive at
20 the input. Word spotting and slot filling may also be used in conjunction with such a PROMPT to determine the user input. For example, a PROMPT may request that the

user speak a date and time, *e.g.*, to choose an airline flight or to make a restaurant reservation. The user's spoken response may be resolved against known date and time formats to determine the input. According to another embodiment, a PROMPT is used to request input using natural language. For instance, in conjunction with a voice service to
5 be used to make travel plans, instead of having separate PROMPT elements request input for flight arrival, departure dates and locations, a single natural language PROMPT may ask, "Please state your travel plan." In response, the user states 'I'd like to go from Washington DC to New York city on the 3rd of January and return on the 3rd of February. This request would be processed using speech recognition and pattern matching
10 technology to derive the user's input.

The ERROR element is used to define the behavior of the system if a user makes an invalid response such as touching a number that has not been associated with an OPTION element, or entering input that does not meet the criteria of a PROMPT element. A SYS-ERROR element defines a handler for certain events, such as expiration
15 of the waiting time for a user response.

The FOR-EACH element is used to direct the system to loop through a list of variables *e.g.*, variables contained in a database report, or variables from a user input, to dynamically generate speech from data.

In addition to the elements described above, there are two features that maximize
20 an administrator's ability to design voice services. Call Flow Reports enable an administrator to generate the structure of a call based on the content of an report *e.g.*,

from an OLAP system or other data repository. For example, the options presented to a user in a PROMPT element may be made to correspond to the row of a data report. According to one embodiment, report data is converted into options by application of an XSL (extensible style sheet language) style sheet. The result of this application is
5 inserted into the static call structure when the voice service is executed.

The use of an XSL style sheet is a feature that maximizes an administrator's voice service building ability. As discussed above, they are used to create dynamic call structure that depends on data report output. They may also be used to generate a text string that comprises the message to be read to a user at any point in a call.

10 A method for creating a voice service according to one embodiment will now be explained in conjunction with Figure 2. The method begins in step 210 by naming the voice service. Then, in step 220 various scheduling parameters of the voice service are defined. In step 250 the service content is defined. And, in step 260, the personalization modes, or style properties are selected for the voice service.

15 According to one embodiment, in step 210, a voice service is named and a description of the voice service provided. By providing a name and description, a voice service may be uniquely identified. An interface is provided for prompting input of the name of the service to be created or edited. An input may also be provided for a written description. An open typing field would be one option for providing the description
20 input. According to another embodiment, if an existing call service has been selected to edit, the service name field may not be present or may not allow modification.

In step 220, conditions for initiating the service are selected. This may include selecting and defining a service type. At least two types of services may be provided based on how the services are triggered. A first type of service is run according to a predetermined schedule and output is generated each time the service is run. A second
5 type of service, an alert service, is one that is run periodically as well, however, output is only generated when certain criteria is satisfied. Other service types may be possible as well. In one embodiment the administrator is prompted to choose between a scheduled service or an alert service. An interface may provide an appropriate prompt and some means for selecting between a scheduled service and an alert service. One option for
10 providing the input might be an interface with a two element toggle list.

In one embodiment, a set of alert conditions is specified to allow the system to evaluate when the service should be initiated if an alert type service has been selected. In one embodiment, a report or a template/filter combination upon which the alert is based is specified. Reports and template/filter combinations may be predefined by other objects
15 in the system including an agent module or object creation module. According to one embodiment, an agent module, such as DSS agent™ offered by MicroStrategy, may be used to create and define reports with filters and template combinations, and to establish the alert criteria for an alert service. According to another embodiment, an interface is be provided which includes a listing of any alert conditions presently selected for the voice
20 service. According to this embodiment, the interface may comprise a display window. A browse feature may take the user to a special browsing interface configured to select a

report or filter-template combination. One embodiment of an interface for selecting reports and filter-template combinations is described below. Once a report or filter and template combination is chosen, the alerts contained in the report or filter and template combination may be listed in the display window of the interface.

53/123 In step 220, the schedule for the service is also selected. According to one embodiment, predefined schedules for voice services may be provided or a customized schedule for the voice service may be created. If a new schedule is to be created, a module may be opened to enable the schedule name and parameters to be set. Schedules may be run on a several-minute, hourly, daily, monthly, semi-annual, annual or other bases, depending upon what frequency is desired. According to one embodiment, an interface is provided that allows the administrator to browse through existing schedules and select an appropriate one. The interface may provide a browsing window for finding existing schedule files and a "new schedule" feature which initiates the schedule generating module. In one embodiment, schedules may not be set for alert type services. However, in some embodiments, a schedule for evaluating whether alert conditions have been met may be established in a similar manner.

SUP 123 In step 220, the duration of the service is also set. Service duration indicates the starting and stopping dates for the service. Setting a service duration may be appropriate regardless of whether a scheduled service or alert type service has been selected. The start date is the base line for the scheduled calculation, while the end date indicates when the voice service will no longer be sent. The service may start immediately or at some

later time. According to one embodiment, interface is provided to allow the administrator to input start and end dates. The interface may also allow the administrator to indicate that the service should start immediately or run indefinitely. Various calendar features may be provided to facilitate selection of start and stop dates. For example, a calendar that specifies a date with pull-down menus that allow selection of a day, month and year may be provided according to known methods of selecting dates in such programs as electronic calendar programs and scheduling programs used in other software products. One specific aid that may be provided is to provide a calendar with a red circle indicating the present date and a blue ellipse around the current numerical date in each subsequent month to more easily allow the user to identify monthly intervals. Other methods may also be used.

In step 220, a voice service may also be designated as a mid-tier slicing service. In one embodiment, mid-tier slicing services generate content and a dynamic subscriber list in a single query to an OLAP system. According to one embodiment, in a mid-tier slicing service a single database query is performed for all subscribers to the service. The result set developed by that query is organized in a table that contains a column that indicates one or more users that each row of data is applicable to.

In step 250, the content of the voice service is defined. Defining the content of the voice service may include selecting the speech to be delivered during the voice service broadcast (content), the structure of dialogs, menus, inputs, and the background procedures which generate both content and structure. In one embodiment, defining

voice service content establishes the procedures performed by the vss server to assemble one or more active voice pages in response to initiation of the voice service. According to one embodiment, defining service content involves establishing a hierarchical structure of TML elements which define the structure and content of a voice service. All of the

5 elements in a given service may be contained within a container.

The personalization type is selected in step 260. Personalization type defines the options that the administrator will have in applying personalization filters to a voice service. According to one embodiment, a personalization filter is a set of style properties that can be used to determine what content generated by the service will be delivered to

10 the individual user and in what format it will be delivered. In one embodiment, personalizing the delivery format may include selection of style properties that determine the sex of the voice, the speed of the voice, the number of call back attempts, etc. Personalization filters may exist for individual users, groups of users, or types of users. According to one embodiment, personalization filters may be created independent of the

15 voice service. According to this embodiment, a voice service specifies what filters are used when generating IVBs. Some personalization type options may include: allowing no personalization filters; allowing personalization filters for some users, but not requiring them; and requiring personalization filters for all interactive voice broadcasts made using the service.

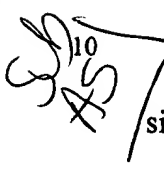
According to one embodiment, specifying personalization type is accomplished by administrator input through an interface. The interface may offer a toggle list with the three options: required personalization, optional personalization, and no personalization.

The voice service may be stored in a database structure to enable users to retrieve
5 predefined voice services and to subscribe to these services, for example, through subscription interfaces explained in conjunction Figures 3a-3c or otherwise. An interface informing the administrator that creation of the voice service is complete may also be provided.

According to one embodiment, the method of Figure 1b also comprises an error
10 condition step. An error condition step may be used to enable administrators to specify “error” conditions and the handling of those conditions. For example, an “error” condition may comprise a notification that a server is “down” or that there is no data to be returned. An administrator may specify particular actions to be performed by the system in response to one or more error conditions. For example, an administrator may specify
15 an “addressing” error (*e.g.*, disconnected number) and indicate a particular action to be performed in response to an “addressing” error (*e.g.*, notify system administrator). Other error conditions might include: an alert report encountering an error and returning no data; a subscriber lacking the required personalization filter for the service; errors occurring in the generation of one or more reports; or reports returning no data. Various
20 other conditions and actions may be specified. Certain error conditions may be predetermined for the system, but an administrator may have reasons for supplementing

or diverging from the predetermined error conditions. According to one particular embodiment, error conditions are specified using the ERROR and SYS-ERROR elements.

In one embodiment, setting error conditions may be accomplished using an error
5 handling interface. The interface may allow the administrator to select either default error handling, or to customize error handling using a module for defining error handling. If default handling is selected, the system uses established settings. If customized handling is chosen, the user may use a feature to access the appropriate interface for the error handling module.

 Servers may have limited capacity to perform all of the actions required of them simultaneously, the method of Figure 1b comprises a step for prioritizing the execution and delivery of voice services. Prioritization may establish the order in which the voice service system allocates resources for processing voice service and delivering the IVB. According to one embodiment, assigning priority to a voice service establishes priority
15 for queries to the database system, formatting the voice service, or IVBs. Any criteria may be used for establishing priority. According to one embodiment, priority is established based on service content. According to another embodiment, priority is based on service destination. According to another embodiment, priority may be established based on the type of voice service, *i.e.*, alert vs. scheduled. Any number of procedures or
20 criteria for denoting relative importance of service delivery may be established.

In one embodiment, an interface is provided for defining the priority of the voice service being created or edited. According to one embodiment, the interface comprises a screen including option boxes with pull down menus listing the number of different prioritization options.

5 Another aspect of the invention relates to a method for executing a voice service. Figure 1c depicts one example of a flow chart for executing a voice service. In step 310, the content of a voice service is generated. In step 320, the call structure of an IVB is created via Active Voice Pages. In step 330, the AVPs are put in a call database for processing *e.g.*, in a call queue. In step 340, the call request is processed and an
10 interactive voice broadcast with the user is implemented. In step 350, user's responses are written back to the voice service system (*e.g.*, the Active Voice Page). Each of these steps will be explained in more detail below.

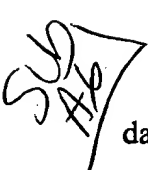
According to one embodiment, content is created in step 310 as follows. A voice service execution begins by running scheduled reports, queries or by taking other action
15 to determine whether the service should be sent. The subscribers for the service are then resolved. Datasets are generated for each group of subscribers that has unique personalization criteria.

Call structure may be created (step 320) as follows. An AVP contains data at various hierarchical content levels (nodes) that can be either static text or dynamic
20 content. Static text can be generated *e.g.*, by typing or by incorporating a text file. Dynamic content may be generated *e.g.*, by inserting data from a data report using a grid

an/or an XSL stylesheet. Moreover, content is not limited to text based information. Other media, such as, sound files, may be incorporated into the AVP. The call data (for example, at a particular level) may be the text that is converted to speech and played when the recipient encounters the node.

5 According to another embodiment, call content may include content from other voice pages, for example, "standard" active voice pages that are generated and inserted into a database or Web Server where the pages are periodically refreshed. According to one particular embodiment, the active voice page that is generated for a user contains links to other active voice pages. The links may be followed using a process similar to
10 web page links.

 The call structure may comprise either a static structure that is defined in the voice service interfaces *e.g.*, by typing text into a text box and/or a dynamic structure generated by grid/XSL combinations. The dynamic structure is merged with static structure during the service execution. A single call structure is created for each group of users that have
15 identical personalization properties across all projects because such a group will receive the same content.

 After a call structure is generated, in step 330, it is sent to a call database *e.g.*, call database 1811 shown in Figure 3 along with the addresses and style properties of the users. The style properties govern the behavior of a call server 18 in various aspects of
20 the dialog with a user. Call server 18 queries call database 1811 for current call requests and places new call requests in its queue.

In step 340, a call request is processed. A call is implemented on call server 18 using one of several ports that are configured to handle telephone communication. When a port becomes available, the call request is removed from the queue and the call is made to the user. As the user navigates through an active voice page, *e.g.*, by entering input using the key pad or by speaking responses, call/content is presented by converting text to speech in text-to-speech engine 1814. User input during the call may be stored for processing. According to another embodiment, user responses and other input may also be used to follow links to other active voice pages. For example, as explained above, “standard” active voice pages may be generated and inserted into a database or Web Server. Then, when a user’s voice service is delivered, that voice service may contain links to information that may be accessed by a user. A user may access those standard active voice pages by entering input in response to OPTION or PROMPT elements.

In step 350, user responses are stored by the system. According to one embodiment, user responses are stored in a response collection defined by the active voice page. A voice service may specify that a subscriber return information during an IVB so that another application may process the data. For instance, a user may be prompted to purchase a commodity and be asked to enter or speak the number of units for the transaction. During or after an IVB, the subscriber’s responses are written to a location from which they can be retrieved for processing (*e.g.*, by an external application).

Fig. 2 is an example of an IVB with interactive call flow. An IVB usually contains a greeting message that addresses the targeted user, identifies the name of the calling application, and states the purpose of the call and/or presents summary metrics. The voice service system can also implement a PIN verification protocol, if this layer of security is required. The main menu structure of an IVB can contain a number of options that lead to sub-menu structures. A menu can also contain prompts for the user to enter numerical information using a telephone touch pad dial. A node along the hierarchical menu structure may have options to return the user to a higher level.

Fig. 3 depicts an embodiment of a system according to one embodiment of the present invention. Preferably, the system comprises database system 12, a DSS server 14, voice service server 16, a call server 18, subscription interface 20, and other input/files 24.

Database system 12 and DSS server 14 comprise an OLAP system that generates user-specified reports from data maintained by database system 12. Database system 12 may comprise any data warehouse or data mart as is known in the art, including a relational database management system ("RDBMS"), a multidimensional database management system ("MDDBMS") or a hybrid system. DSS server 14 may comprise an OLAP server system for accessing and managing data stored in database system 12. DSS server 14 may comprise a ROLAP engine, MOLAP engine or a HOLAP engine according to different embodiments. Specifically, DSS server 14 may comprise a multithreaded server for performing analysis directly against database system 12.

According to one embodiment, DSS server 14 comprises a ROLAP engine known as DSS Server™ offered by MicroStrategy.

Voice service server (VSS) 16, call server 18 and subscription interface 20 comprise a system through which subscribers request data and reports *e.g.*, OLAP reports through a variety of ways and are verbally provided with their results through an IVB. During an IVB, subscribers receive their requested information and may make follow-up requests and receive responses in real-time as described above. Although the system is shown, and will be explained, as being comprised of separate components and modules, it should be understood that the components and modules may be combined or further separated. Various functions and features may be combined or separated

Subscription interface 20 enables users or administrators of the system to monitor and update subscriptions to various services provided through VSS 16. Subscription interface 20 includes a world wide web (WWW) interface 201, a telephone interface 202, other interfaces as desired and a subscriber API 203. WWW interface 201 and telephone interface 202 enable system 100 to be accessed, for example, to subscribe to voice services or to modify existing voice services. Other interfaces may be used. Subscriber API 203 provides communication between subscription interface 20 and VSS 16 so that information entered through subscription interface 20 is passed through to VSS 16.

Subscription interface 20 is also used to create a subscriber list by adding one or more subscribers to a service. Users or system administrators having access to VSS 16 may add multiple types of subscribers to a service such as a subscriber from either a static

recipient list (SRL) (*e.g.*, addresses and groups) or a dynamic recipient list (DRL) (described in further detail below). The subscribers may be identified, for example, individually, in groups, or as dynamic subscribers in a DRL. Subscription interface 20 permits a user to specify particular criteria (*e.g.*, filters, metrics, etc.) by accessing
5 database system 12 and providing the user with a list of available filters, metrics, etc. The user may then select the criteria desired to be used for the service. Metadata may be used to increase the efficiency of the system.

A SRL is a list of manually entered names of subscribers of a particular service. The list may be entered using subscription interface 20 or administrator console 161.
10 SRL entries may be personalized such that for any service, a personalization filter (other than a default filter) may be specified. A SRL enables different personalizations to apply for a login alias as well. For example, a login alias may be created using personalization engine 1632. Personalization engine 1632 enables subscribers to set preferred formats, arrangements, etc. for receiving content. The login alias may be used to determine a
15 subscriber's preferences and generate service content according to the subscriber's preferences when generating service content for a particular subscriber.

A DRL may be a report which returns lists of valid user names based on predetermined criteria that are applied to the contents of a database such as database system 12. Providing a DRL as a report enables the DRL to incorporate any filtering
20 criteria desired, thereby allowing a list of subscribers to be derived by an application of a filter to the data in database system 12. In this manner, subscribers of a service may be

altered simply by changing the filter criteria so that different user names are returned for the DRL. Similarly, subscription lists may be changed by manipulating the filter without requiring interaction with administrator console 161. Additionally, categorization of each subscriber may be performed in numerous ways. For example, subscribers may be
5 grouped via agent filters. In one specific embodiment, a DRL is created using DSS Agent™ offered by MicroStrategy.

VSS 16 is shown in more detail in Figure 3b. According to one embodiment, VSS 16 comprises administrator console 161, voice service API 162 and backend server 163. Administrator console 161 is the main interface of system 100 and is used to view
10 and organize objects used for voice broadcasting. Administrator console 161 provides access to a hierarchy of additional interfaces through which a system administrator can utilize and maintain system 100. Administrator console 161 comprises system administrator module 1611, scheduling module 1612, exceptions module 1613, call settings module 1614, address handling module 1615, and service wizard 1616.

15 System administrator module 1611 comprises a number of interfaces that enable selection and control of the parameters of system 100. For example, system administrator module 1611 enables an administrator to specify and/or modify an email system, supporting servers and a repository server with which system 100 is to be used. System administrator 1611 also enables overall control of system 100. For example, system
20 administrator module is also used to control the installation process and to start, stop or

idle system 100. According to one embodiment, system administrator 1611 comprises one or more graphical user interfaces (GUIs).

Scheduling module 1612 comprises a number of interfaces that enable scheduling of voice services. Voice services may be scheduled according to any suitable methodology, such as according to scheduled times or when a predetermined condition is met. For example, the predetermined condition may be a scheduled event (time-based) including, day, date and/or time, or if certain conditions are met. In any event, when a predetermined condition is met for a given service, system 100 automatically initiates a call to the subscribers of that service. According to one embodiment, scheduling module 1612 comprises one or more GUIs.

Exceptions module 1613 comprises one or more interfaces that enable the system administrator to define one or more exceptions, triggers or other conditions. According to one embodiment, exceptions module 1613 comprises one or more GUIs.

Call settings module 1614 comprises one or more interfaces that enable the system administrator to select a set of style properties for a particular user or group of users. Each particular user may have different options for delivery of voice services depending on the hardware over which their voice services are to be delivered and depending on their own preferences. As an example of how the delivery of voice services depends on a user's hardware, the system may deliver voice services differently depending on whether the user's terminal device has voice mail or not. As an example of how the delivery of voice services depends on a user's preferences, a user may chose

to have the pitch of the voice, the speed of the voice or the sex of the voice varied depending on their personal preferences. According to one embodiment, call settings module 1614 comprises one or more GUIs.

Address handling module 1615 comprises one or more interface that enable a
5 system administrator to control the address (e.g., the telephone number) where voice services content is to be delivered. The may be set by the system administrator using address handling module 1615. According to one embodiment, address handling module 1615 comprises one or more GUIs.

Voice service wizard module 1616 comprises a collection of interfaces that enable
10 a system administrator to create and/or modify voice services. According to one embodiment, service wizard module 1616 comprises a collection of interfaces that enable a system administrator to define a series of dialogs that contain messages and inputs and determine the call flow between these dialogs based on selections made by the user. The arrangement of the messages and prompts and the flow between them comprises the
15 structure of a voice service. The substance of the messages and prompts is the content of a voice service. The structure and content are defined using service wizard module 1616.

Voice service API 162 (e.g., MicroStrategy Telecaster Server API) provides communication between administrator console 161 and backend server 163. Voice Service API 162 thus enables information entered through administrator console 161 to
20 be accessed by backend server 163 (e.g., MicroStrategy Telecaster Server).

Backend server 163 utilizes the information input through administrator console 161 to initiate and construct voice services for delivery to a user. Backend server 163 comprises report formatter 1631, personalization engine 1632, scheduler 1633 and SQL engine 1634. According to one embodiment, backend server 163 comprises

5 MicroStrategy Broadcast Server. Report formatter 1631, personalization engine 1632, and scheduler 1633 operate together, utilizing the parameters entered through administrator console 161, to initiate and assemble voice services for transmission through call server 18. Specifically, scheduler 1633 monitors the voice service schedules and initiates voice services at the appropriate time. Personalization engine 1632 and

10 report formatter 1631 use information entered through service wizard 1616, exceptions module 1613, call settings module 1614, and address module 1615, and output provided by DSS server 14 to assemble and address personalized reports that can be sent to call server 18 for transmission. According to one embodiment, report formatter 1631 includes an XML based markup language engine to assemble the voice services. In a particular

15 embodiment, report formatter includes a Telecaster Markup Language engine offered by MicroStrategy Inc. to assemble the call content and structure for call server 18.

SQL engine 1634 is used to make queries against a database when generating reports. More specifically, SQL engine 1634 converts requests for information into SQL statements to query a database.

20 Repository 164 may be a group of relational tables stored in a database. Repository 164 stores objects which are needed by system 100 to function correctly.

More than one repository can exist, but preferably the system 100 is connected to only one repository at a time.

According to one embodiment, a call server 18 is used to accomplish transmission of the voice services over standard telephone lines. Call server 18 is shown in more detail in Figure 3c. According to one embodiment, call server 18 comprises software components 181 and hardware components 182. Software components 181 comprise call database 1811, mark-up language parsing engine 1812, call builder 1813, text-to-speech engine 1814, response storage device 1815 and statistic accumulator 1816.

Call database 1811 comprises storage for voice services that have been assembled in VSS 16 and are awaiting transmission by call server 18. These voice services may include those awaiting an initial attempt at transmission and those that were unsuccessfully transmitted (*e.g.*, because of a busy signal) and are awaiting re-transmission. According to one embodiment, call database 1811 comprises any type of relational database having the size sufficient to store an outgoing voice service queue depending on the application. Call database 1811 also comprises storage space for a log of calls that have been completed.

Voice services stored in call database 1811 are preferably stored in a mark-up language. Mark-up language parsing engine 1812 accepts these stored voice services and separates the voice services into parts. That is, the mark-up language version of these voice services comprises call content elements, call structure elements and mark-up

language instructions. Mark-up language parsing engine 1812 extracts the content and structure from the mark-up language and passes them to call builder 1813.

Call builder 1813 is the module that initiates and conducts the telephone call to a user. More specifically, call builder dials and establishes a connection with a user and passes user input through to markup language parsing engine 1812. In one embodiment, call builder 1813 comprises "Call Builder" software available from Call Technologies Inc. Call builder 1813 may be used for device detection, line monitoring for user input, call session management, potentially transfer of call to another line, termination of a call, and other functions.

Text-to-speech engine 1814 works in conjunction with mark-up language parsing engine 1812 and call builder 1813 to provide verbal communication with a user. Specifically, after call builder 1813 establishes a connection with a user, text-to-speech engine 1814 dynamically converts the content from mark-up language parsing engine 1812 to speech in real time.

A voice recognition module may be used to provide voice recognition functionality for call server 181. Voice recognition functionality may be used to identify the user at the beginning of a call to help ensure that voice services are not presented to an unauthorized user or to identify if a human or machine answers the call. This module may be a part of call builder 1813. This module may also be used to recognize spoken input (say "one" instead of press "1"), enhanced command execution (user could say "transfer money from my checking to savings"), enhanced filtering (instead of typing


stock symbols, a user would say "MSTR"), enhanced prompting, (saying numeral values).

User response module 1815 comprises a module that stores user responses and passes them back to intelligence server 16. Preferably, this is done within an AVP.


5 During a telephone call, a user may be prompted to make choices in response to prompts by the system. Depending on the nature of the call, these responses may comprise, for example, instructions to buy or sell stock, to replenish inventory, or to buy or rebook an airline flight. User response module 1815 comprises a database to store these responses along with an identification of the call in which they were given. The identification of
10 the call in which they were given is important to determining what should be done with these responses after the call is terminated. User responses may be passed back to intelligence server 16 after the call is complete. The responses may be processed during or after the call, by the system or by being passed to another application.

Statistics accumulator 1816 comprises a module that accumulates statistics
15 regarding calls placed by call builder 1813. These statistics including, for example, the number of times a particular call has been attempted, the number of times a particular call has resulted in voice mail, the number of times a user responds to a call and other statistics, can be used to modify future call attempts to a particular user or the structure of a voice service provided to a particular user. For example, according to one embodiment,
20 statistics accumulator 1816 accumulates the number of times a call has been unsuccessfully attempted by call builder 1813. This type of information is then used by

call server 18 to determine whether or not the call should be attempted again, and whether or not a voice mail should be left.

5  Call server 18 also comprises certain hardware components 182. As shown in Figure 1c, hardware components 182 comprise processor 1821 and computer telephone module 1822. According to one embodiment, processor 1821 comprises a Pentium II processor, available from Intel, Inc. Module 1822 provides voice synthesis functionality that is used in conjunction with Text to Speech engine 1814 to communicate the content of voice services to a user. Module 1822 preferably comprises voice boards available from Dialogic, Inc. Other processors and voice synthesizers meeting system requirements may be used.

The system and method of the present invention may form an integral part of an overall commercial transaction processing system.

15  According to one embodiment of the present invention, a system and method that enable closed-loop transaction processing are provided. The method begins with the deployment of an IVB by executing a service. As detailed above, this includes generating the content and combining this with personalization information to create an active voice page. Call server 18 places a call to the user. During the call, information is delivered to the user through a voice-enabled terminal device (e.g., a telephone or cellular phone).

20 During the IVB, a user may request a transaction, service, further information from the database or other request, e.g., based on options presented to the user. These will generically be referred to as transactions. The request may be, but is not necessarily,

based on or related to information that was delivered to the user. According to one embodiment, the request comprises a user response to a set of options and/or input of information through a telephone keypad, voice input or other input mechanism.

According to another embodiment, the request can be made by a user by speaking the

5 request. Other types of requests are possible.

According to one embodiment, the user responses are written to a response collection, which along with information stored in the active voice page, can be used to cause a selected transaction to be executed. According to one embodiment, the active voice page comprises an XML-based document that includes embedded, generic requests, e.g., a request for a transaction, or a request for additional information (a database query). These embedded requests are linked with, for example option statements or prompts so that when a user enters information, the information is entered into the generic request and thus completes a specific transaction request. For example, in the example if a user exercises an option to buy a particular stock, that stock's ticker symbol is used to

10 complete a generic "stock buy" that was embedded in the active voice page.

15

According to one embodiment, tokens are used to manage user inputs during the IVB. A token is a temporary variable that can hold different values during an IVB.

When a user enters input, it is stored as a token. The token value is used to complete a transaction request as described above. According to one embodiment, the system

20 maintains a running list of tokens, or a response collection, during an IVB.

In order to complete the requested transaction, the user responses (and other information from the active voice page) may need to be converted to a particular format. The format will depend, for example, on the nature and type of transaction requested and the system or application that will execute the transaction. For example, a request to
5 purchase goods through a web-site may require the information to be in HTML/HTTP format. A request for additional information may require and SQL statement. A telephone-based transaction may require another format.

Therefore, the transaction request is formatted. According to one embodiment, the transaction is formatted to be made against a web-based transaction system.

10 According to another embodiment, the transaction request is formatted to be made against a database. According to another embodiment, the transaction is formatted to be made against a telephone-based transaction system. According to another embodiment, the transaction is formatted to be made via e-mail or EDI. Other embodiments are possible.

In one embodiment, the formatted transaction request comprises an embedded
15 transaction request. The system described in connection with Figures 1-3 provides interactive voice services using TML, a markup language based on XML. Using TML active voice pages are constructed that contain the structure and content for a interactive voice broadcast including, inter alia, presenting the user with options and prompting the user for information. Moreover in connection with OPTION and PROMPT elements,
20 active voice pages also can include embedded statements such as transaction requests.

Therefore, the formatting for the transaction request can be accomplished ahead of time based on the particular types of transactions the user may select.

For example, in connection with an exemplary stock purchase, an active voice page can include an embedded transaction request to sell stock in the format necessary for a particular preferred brokerage. The embedded statement would include predefined variables for the name of the stock, the number of shares, the type of order (market or limit, etc.), and other variables. When the user chooses to exercise the option to buy or sell stock, the predefined variables are replaced with information entered by the user in response to OPTION or PROMPT elements. Thus, a properly formatted transaction request is completed.

In the system of Figures 1-3, TML parsing engine in call server 18 includes the functionality necessary to generate the properly formatted transaction request as described above. For example, in connection with the embodiment described above, the TML parsing engine shown in Figure 3c reads the active voice pages. When the TML parsing engine reads an OPTION element that includes an embedded transaction request, it stores the transaction request, and defines the necessary variables and variable locations. When the user exercises that OPTION, the user's input is received by the TML parsing engine and placed at the memory locations to complete the transaction request. This technique could be used, for example, to generate a formatted transaction request for a web-site.

According to another embodiment, where the transaction request is made via a natural language, voice request, a formatted transaction request can be generated in a number of ways. According to one embodiment, speech recognition technology is used to translate the user's request into text and parse out the response information. The text is
5 then used to complete an embedded transaction request as described above. According to another embodiment, speech recognition software is used to translate the request to text. The text is then converted to a formatted request based on a set of known preferences.

A connection is established with the transaction processing system. This can be accomplished during, or after the IVB. According to one embodiment, the transaction
10 processing system comprises a remotely located telephone-based transaction site. For example, in the system shown in Figures 1-3, call server 18, through the TML parsing engine 1812, establishes a connection with a telephone-based transaction processing site.

According to another embodiment, the transaction processing system comprises a remotely based web-site. According to this embodiment, the formatted request includes a
15 URL to locate the web-site and the system accesses the site through a web connection using the formatted request. Alternatively, the formatted request includes an e-mail address and the system uses any known email program to generate an e-mail request for the transaction.

After the connection is established, the transaction is processed by the transaction
20 processing site and the user is notified of the status of the transaction. If the transaction is completed in real-time, the user may be immediately notified. If the transaction is

executed after the IVB, the user may be called again by the system, sent an e-mail, or otherwise notified when the transaction has been completed.

According to one particular embodiment, the system comprises the interactive voice broadcasting system shown and described in Figures 1-3 and the transaction is accomplished in real-time. In this embodiment, confirmation of the transaction is returned to TML parsing engine 1812 shown in Figure 3 and translated to speech in text-to-speech engine 1814 and presented to the user during the IVB. More specifically, and similar to the process described with respect to embedded formatted transaction requests, TML also enables embedding of a response statement. Thus, when the transaction is processed and confirmation of the transaction is returned to the system, an embedded confirmation statement is conveyed to the user through TML parsing engine 1812 after being converted to speech in text-to-speech engine 1814.

Figure 4 schematically depicts one example of how the system and method of the present invention would fit into such a commercial transaction processing system. Working from left to right in Figure 4, the system begins and ends with information stored in relational databases. One of the primary purposes of information is in making decisions. Thus, the information in the databases is most useful if provided to someone who desires it in a timely fashion.

A voice service system is provided to enable access to the information in the databases. The voice service system utilizes personalization information and personalized menus to construct AVPs pages that enable the information to be delivered

to a user verbally. Moreover, the AVPs pages, not only enable information to be presented to the user. But, they also enable the user to provide information back to the voice service system for additional processing.

According to the embodiment shown in Figure 4, once the AVPs are constructed
5 by voice service system, they are processed and the content is delivered to a user verbally in an IVB. Thus, call processing and text-to-speech technology are used to establish a telephone connection with a user and convert the active voice pages to speech for presentation to the user. As shown in Figure 4, the IVB may be delivered to a user in many devices, including a telephone, a mobile phone, voice mail, an answering machine
10 or any other voice-enabled device.

During the IVB, depending on the content that is being delivered, control may be passed to an e-commerce application for the user to complete a transaction based on the information presented. For example, if the user has requested information about sales on a particular brand of merchandise, the user may be connected with a particular retailer in
15 order to complete a transaction to buy a particular good or service. Information about this transaction is then added to the databases and thus may be advantageously accessed by other users.

It may not be economical for some potential users of a voice broadcasting system to buy and/or maintain their own telephony hardware and software as embodied in call
20 server 18. In such a case, a voice service bureau may be maintained at a remote location to service users voice service requests. A voice service bureau and a method of using a

voice service bureau according to various embodiments of the present invention is described in conjunction with Figures 5-6.

In one embodiment, a voice service bureau may comprise one or more call servers and call databases that are centrally located and enable other voice service systems to generate a call request and pass the call request to the VSB to execute a call. In this way the other voice service systems do not need to invest in acquiring and maintaining call data bases, call servers, additional telephone lines and other equipment or software. Moreover, the VSB facilitates weeding out usage of illegal numbers and spamming by number checking implemented through its web server.

A voice service bureau and a method of using a voice service bureau according to one embodiment are described in conjunction with Figures 5-6. Figure 5 depicts a method of utilizing a voice service bureau according to one embodiment of the present invention. The method begins in step 810 with a request to place one or more telephone calls received through a computer network.

According to one embodiment, the voice service bureau is maintained at a location distant from the voice service system. Therefore, in order for a voice service to be processed by the voice service bureau, in step 810 the voice service is sent to the voice services bureau, preferably over some secure line of communication. According to one embodiment, the request is sent to the voice service bureau through the Internet using secure HTTPS. HTTPS provides a secure exchange of data between clients and the voice service bureau using asymmetric encryption keys based on secure server certificates. In

another embodiment, SSL HTTP protocol is used to send a call request to the voice service bureau. Both of these protocols help ensure that a secure channel of communication is maintained between the voice service system and the voice service bureau. Other security techniques may be used.

5 When a request for a call or IVB is received, by the VSB, the request is authenticated by the voice service bureau in step 820. According to one embodiment, the authenticity of the request is determined in at least two ways. First, it is determined whether or not the request was submitted from a server having a valid, active server certificate. More specifically, requests may be typically received via a stream of HTTPS
10 data. Each such request originating from a server with a valid server certificate will include an embedded code (i.e., server certificate) that indicates the request is authentic. In addition to the use of server certificates, each request may also be authenticated using an identification number and password. Therefore, if the request submitted does not include a valid server certificate and does not identify a valid I.D./password combination,
15 the request will not be processed. The step of authenticating also comprises performing any necessary decryption. According to one embodiment, any errors that are encountered in the process of decrypting or authenticating the call request are logged an error system and may be sent back to the administrator of the sending system. Other methods of authenticating the request are possible.

20 Each properly authenticated request is sent to a call server (step 830) and processed (step 840). According to one embodiment, the voice service bureau comprises


a number of call servers. According to one embodiment, the calls are sent to a call database, and processed as set forth herein in conjunction with the explanation of call server 18.

One embodiment of a voice service bureau will now be explained in conjunction with Figures 6a-6c. Figure 6a depicts a system comprising a plurality of client side installations 91, a primary voice bureau 92, a system administrator 93, a backup voice service bureau 94, and a plurality of users 95. Client side installations 91 communicate with voice service bureau 92 through a computer network. Voice service bureau 92 communicates with users 95 through a voice network. According to one embodiment, the computer network comprises the internet and client side installations 91 communicate with voice service bureau 92 using HTTPS as described above, and the voice network comprises a public telephone network.

According to one embodiment, client side installations 91 are substantially identical to the system shown in Figure 4 except for the elimination of call server 18. In the system of Fig. 6a, the functionality of call server 18 is performed by VSB 92. As shown in this embodiment, VSB 92 can service multiple client side installations 91₁ to 91_n. According to another embodiment, client-side installation functionality may be included within VSB 92. According to this embodiment VSB 92 constitutes a fully functional voice service that is accessible through email, telephone or other interfaces.

According to this embodiment, when voice services have been assembled by intelligence server 16, a request to have the voice services transmitted is sent via a secure

network connection through the computer network shown to primary voice bureau 92 and backup voice service bureau 94 as described above. According to one embodiment, the request comprises a mark-up language string that contains the voice service structure and content and personal style properties and other information. As described above, voice bureau 92 authenticates the request, queues the voice services and sends IVBs to users 95 through the voice network.

 A block diagram of one embodiment of primary voice bureau 92 is shown in Figure 6b. According to this embodiment, primary voice bureau comprises routers 921, dual-homed servers 922, database servers 923, call database 924, backup storage 925, call servers 926, internal switch 927, and system administrator 928. Routers 921 receive call requests via a computer network and pass them along to one of the two dual-homed servers 922. Router 921 monitors activity on servers 922 and forwards call requests to one of the two depending on availability.

Dual-homed servers 922 comprise servers configured to receive and send HTTPS email. As part of their receiving function, dual-homed servers 922 are configured to perform the authentication processing described above. According to one embodiment, dual-homed servers 922 determine whether the incoming request originated from a server with an active server certificate and also determine if the request contains a valid I.D./password combination. Once dual-homed servers 922 have authenticated the incoming request, they forward the request to be queued in call database 924. As part of their sending function, dual-homed servers 922 are configured to format and send HTTPS

email. As discussed above, during an IVB a user may request that further information be accessed from a database or that some transaction be performed. According to one embodiment, these user requests are forwarded back to the originating system via HTTPS email by dual-homed servers 922. Dual-homed servers 922 are load balanced to facilitate
5 optimal performance and handling of incoming call requests.

Database servers 923, call database 924, and backup storage 925 together comprise a call request queuing system. Primary voice bureau 92 is configured to handle a large number of call requests. It may not be possible to process call requests as they arrive. Therefore, call requests are queued in call database 924. According to one
10 embodiment, call database 924 comprises a relational database that maintains a queue of all call requests that need to be processed as well as logs of calls that have been processed. According to another embodiment, primary VSB 92 may include a failover measure that enables another system server to become the call database if call database 924 should fail.

15 Database servers 923 are configured to control access to call database 924. According to one embodiment, database servers may be optimized to generate SQL statements to access entries in call database at high speed. Database servers 923 also control storage of call requests and call logs in call database 924.

Call servers 926 each are configured to format and send IVBs. According to one
20 embodiment, each of call servers 926 is substantially identical to call server 18 shown in Figure 3c. More specifically, each of call servers 926 receives requests for IVBs, parses

the call content from the mark-language, establishes a connection with the user through phone lines 929, and receives user responses. According to one embodiment, call servers 926 comprise a clustered architecture that facilitates message recovery in the event of server failure.

5 Primary voice bureau 92 is controlled by system administrator 93 and internal switch 927. System administrator controls switch 927 and thus controls the flow of call requests to call database 924 from dual homed servers 922 and to call servers 926 from call database 924.

 System administrator 93 is also configured to perform a number of other services
10 for primary voice bureau 92. According to one embodiment, system administrator 93 also comprises a billing module, a statistics module, a service module and a security module. The billing module tabulates the number of voice service requests that come from a particular user and considers the billing plan that the customer uses so that the user may be appropriately billed for the use of voice bureau 92. The statistics module
15 determines and maintains statistics about the number of call requests that are processed by voice bureau 92 and statistics regarding call completion such as, e.g., success, failed due to busy signal and failed due to invalid number. These statistics may be used, for example, to evaluate hardware requirements and modify pricing schemes. The security module monitors activity on voice bureau 92 to determine whether or not any
20 unauthorized user has accessed or attempted to access the system. The service module provides an interface through which primary voice bureau 92 may be monitored, for

example, to determine the status of call requests. Other service modules are possible. Moreover, although these services are described as distinct modules, their functionality could be combined and provided in a single module.

5 Backup voice service bureau 94 receives a redundant request for voice services. Backup voice service bureau 94 processes the requests only when primary voice service bureau is offline or busy. One embodiment of backup voice service bureau 94 is shown in Figure 6c. Backup voice bureau 94 comprises routers 941, HTTP server 942, database server 943, call server 946 and routers 947. Each of these components performs a function identical to the corresponding element in primary voice bureau 92. Router 947
10 replaces switch 927. Router 947 controls the forwarding of call requests to database server 943 for queuing in an internal database, and the forwarding of call requests to call server 946 from database server 943.

The systems and methods discussed above are directed to outbound broadcasting of voice services. Nevertheless, in certain situations, for example when the out bound
15 IVB is missed, it is desirable to for a voice service system to enable inbound calling. According to another embodiment, a method and system for providing integrated inbound and outbound voice services is disclosed.

A method for providing inbound access to voice services according to one embodiment of the present invention is shown in Figure 7. According to Figure 7, the
20 method begins with receipt of a call requesting voice services in step 1210. To help ensure system integrity and to prevent unauthorized access, a call request is authenticated

in step 1220. According to one embodiment, each incoming caller is automatically prompted to enter a login identifier and a PIN. According to another embodiment, an automatic number identification system is used, in addition to a login identifier and PIN system, to determine whether or not the user is calling from an authorized device.

5 According to another embodiment, speaker recognition technology is utilized to identify a caller. According to this embodiment, voice prints for each user of the voice service system are stored as identifiers. When an inbound call is connected, pattern matching techniques are used verify the user's speech against the previously stored voice prints. Other security measures are possible.

10 In step 1230, a voice page is located. As explained above, an IVB of a voice service is driven by an active voice page. Accordingly, a user calling in to access voice services locates the desired active voice page. According to one embodiment, the user is automatically placed into an active voice page of a voice service that the user missed. That is, the system chooses an active voice page that it was unable to deliver. According
15 to this embodiment, when a call is undeliverable (e.g., when an answering machine picks up), the active voice page for that call is placed in memory in a "voice site" table or as an active voice page on a web site and addressed using the user's identification. When the user calls in to retrieve the voice service, after the user logs in, the table or web site will be searched for an active voice page that corresponds to their identification. If such a
20 page exists, it is executed by the call server.

Other possibilities exist for accessing active voice pages through inbound calling. According to another embodiment, the system maintains a log of all voice services sent and provides an inbound user an option to select one of their previous voice services. According to another embodiment, an inbound caller is automatically placed into an active voice page that presents the user with an option to select one of that user's most frequently used services. According to still another embodiment, the user is allowed to search for past active voice pages by date or content. For example, the user may be prompted to enter a date on or near which the desired voice page was executed. According to another embodiment, the user may use the telephone keys to enter a search term and search the content of any previously executed active voice page that they are authorized to access or that is not secure.

Once an active voice page is located, the user navigates through the active voice page in step 1240. As described above, a user navigates through an active voice by exercising options, responding to prompts and otherwise entering input to the system. An inbound calling system would thus have access to the full functionality of the voice service system described in conjunction with Figures 1-6.

Figure 8 depicts a block diagram of a call server 18a that enables integrated inbound and outbound calling. In addition to the modules depicted in call server 18 of Figure 3, call server 18a comprises call receiver module 1817, security module 1818 and search module 1819. Moreover, in the system for permitting inbound and outbound calling, call database 1811 has been replaced with an enhanced call database 1811a.

In order to receive inbound calls, call server 18a comprises call receiver module 1817. Although, call server 18 discussed above contains hardware permitting reception of calls as well as transmission of calls, it is not set up to receive calls. Call receiver module 1817 enables call server 18a to receive calls and routes the incoming calls to security module 1818. According to one embodiment, call receiver module comprises a software component designed to configure call server 18a to receive calls. Other embodiments are possible.

Received calls are forwarded to security module 1818 for authentication. According to one embodiment discussed above, incoming calls are authenticated using login I.D.'s and passwords. According to another embodiment, automatic number identification software is used to identify and authenticate callers. According to another embodiment, speech recognition and pattern matching techniques are used to identify a caller.

Authenticated calls may search for an active voice page using search module 1819. According to one embodiment, search module 1819 comprises a search engine designed specifically to search active voice pages. According to one embodiment discussed above, active voice pages utilize an XML-based language and search module 1819 comprises an XML-based search engine. According to another embodiment, search module 1819 comprises a SQL engine designed to make queries against a relational or other type of database.

The active voice pages that are being search are stored in enhanced call database 1811a. In addition to its facilities to queue and log calls, enhanced call database 1811 includes facilities to catalog active voice pages. According to one embodiment, enhanced call database comprises a relational or other type of database. According to this
5 embodiment, enhanced call database is used to store and categorize active voice pages and corresponding parameters, such as expiration dates for active voice pages. Other storage facilities are possible.

Various features and functions of the present invention extend the capabilities of previously known information delivery systems. One such system is MicroStrategy's
10 Broadcaster version 5.6. The features and functions of the present invention are usable in conjunction with Broadcaster and other information delivery systems or alone. Other products may be used with the various features and functions of the invention including, but not limited to, MicroStrategy's known product suite.

Other embodiments and uses of the invention will be apparent to those skilled in
15 the art from consideration of the specification and practice of the invention disclosed herein. The specification and examples should be considered exemplary only. The scope of the invention is only limited by the claims appended hereto.